

VoIP 네트워크에서 패킷 전송지연시간 변이현상을 없애주는 적응식 변이 제어기 제안 및 성능분석

정회원 정 윤 찬*, 조 한 민**

Design of Jitter elimination controller for concealing interarrival packet delay variation in VoIP Network

Yoon-Chan Jung*, Han-Min Jo** *Regular Members*

요 약

이 논문에서는 수신측에 도달하는 VoIP 패킷들을 셰이핑하고 또 버퍼링하기 위하여 CAM형의 제어기를 사용하는 기술로 적응식 셰이핑 제어기를 제안한다. 패킷간 도착 시간 간격의 변이를 없애기 위하여, 지연 균등화 방법에 입각한 전통적인 jitter buffer는 요구되는 버퍼크기가 너무 커지고 이에 기인한 지나친 지연으로 오디오 질이 나빠지는 원인이다. 그러나, 이 논문에서 제안한 방법을 사용하면, 도착시간 간격 변이를 없애기 위한 셰이핑에 의해 야기된 지연은 동적으로 IP 네트워크에서 존재하는 jitter 수준의 정도에 따라 증가 또는 감소한다. 이것은 네트워크의 지연시간 변이 상황에 적응하여 동작하게 되는데, 네트워크에 지연시간 변이현상이 심하지 않으면 셰이핑 제어기에서의 지연시간이 작아진다는 것을 의미한다. 또 CAM형으로 동작하기 때문에 셰이핑 과정이 빨리 이루어지고 이는 수신측에서의 VoIP 패킷 처리 시간을 최대한 단축시켜 준다. 마지막으로 셰이핑 제어기에서의 평균 TALKratio와 jitter buffer 엔트리 수에 대한 패킷 손실과 지연 성능의 관계를 분석한다. 놀랍게도, 제안한 셰이핑 제어기를 사용할 때의 평균 지연이 약 10msec라는 것을 확인하였다. 이 성능은 수신측에서 60msec 강제지연을 강요하는 지연시간 균등화 방법보다 훨씬 더 좋은 성능을 보여주는 것이다.

ABSTRACT

We propose an adaptive shaping controller equipped with the technologies of shaping and buffering VoIP packets arriving at the receiving end by the CAM-type controller. In order to conceal interarrival packet delay variation, the conventional jitter buffers force them to be too large, thereby causing the audio quality to suffer excessive delay. However, by using our proposed method, the delay caused by shaping operation dynamically increases or decreases on the level of jitter that exists with in the IP network. This makes the delay accommodates adaptively the network jitter condition. The less jitter network has the fewer delay the shaping controller requires for jitter elimination. And the CAM-type method generally makes the shaping operation faster and leads to processing packets in as little time as can. We analyze the packet loss and delay performance dependency on the average talk ratio and the number of jitter buffer entries in shaping controller. Surprising, we show that the average delay using our shaping controller is about 10msec. This performance is much better than with the delay equalization method which forces the receiving end to delay about 60msec.

* 가톨릭대학교 컴퓨터전자공학부 (ycjung@www.cuk.ac.kr),
논문번호 : K01178-0810, 접수일자 : 2001년 8월 10일

** (주)모텍스 (nickyjo@shinbiro.com)

I. 서론

일반적으로 IP(Internet Protocol) 네트워크인 인터넷을 통하여 음성과 영상을 전송시키기 위하여 ITU-T 표준인 H.323을 사용한다. H.323 표준은 호 시그널링과 제어, 멀티미디어 전달과 처리, 그리고 대역폭 제어에 관한 사항을 언급하고 있다. 그림 1.1과 같이 H.323 프로토콜은 여러 개의 프로토콜들로 복합 구성된다^[1].

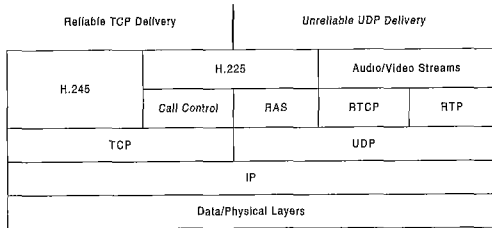


그림 1.1 H.323 프로토콜 슈트 계층 구조

H.323 프로토콜을 사용하는 통신에서 호의 제어 절차는 H.225 권고안을 따른다. 이 H.225는 Q.931로 표현되는 시그널링 메시지의 사용을 전제로 하고 있으며 신뢰성 있는 통신을 위해 TCP를 이용하고 H.225방식의 호 제어 채널 하나가 TCP 포트 1720번 위에서 동작한다^{[1],[2]}. 이 포트를 통하여 IP 네트워크 양쪽 끝에 있는 두 단말 사이의 호를 접속하고, 호를 유지하고 또 호를 해제하는 Q.931 호 제어 메시지가 H.225 호 제어 절차에 따라 이 TCP 포트를 통하여 송·수신되게 된다.

일단 송·수신 단말간에 호 접속이 이루어지면 RTP는 통신 미디어의 전달을 담당하게 된다. 바로 이 RTP가 IP 네트워크 상에서의 대화형 음성, 비디오, 및 데이터 정보를 단말에서 단말까지 실시간으로 전달 가능케 하는 프로토콜이다. 여기서 RTP가 QoS를 어떻게 잘 보장하느냐 하는 문제는 전적으로 RTP 아래쪽에 있는 하위 계층 프로토콜에 달려 있다. 또 H.323을 이용하는 멀티미디어 통신에서 미디어의 전달에서는 UDP 전달 프로토콜을 사용하기 때문에 UDP 전달 서비스 특성과 IP 서비스 특성이 합해진 통신 품질 특성 형태가 H.323을 이용하는 멀티미디어 통신의 QoS로 나타나게 된다^{[1],[3],[4]}.

VoIP 전화에서 나타나는 VoIP 폰 음질에 영향

을 미치는 중요한 요소 가운데 하나는 패킷간 도착 시간 간격의 변이 현상(Jitter)이다. 이 도착시간 변이 현상은 호마다 전용의 경로가 동기식 링크들로 구성 운영되고 또 TSI(Time Slot Interchange)기능으로 스위칭이 “Bypass”식으로 이루어지는 음성전화망에서는 존재하지 않고 오직 패킷 스위칭방식에 기반한 인터넷에서 발생한다. IP 네트워크를 통하여 패킷 음성을 보내는 입장에서는 음성 패킷들을 일정한 간격마다 보내게 된다. 그런데 이 음성 패킷들이 인터넷을 통하여 수신지점에 도착할 때는 일정한 간격의 패킷 흐름이 아니고 패킷이 송신 지점에서 출발하여 수신 지점에 도착할 때까지의 시간 지연이 패킷 각각마다 다르게 나타난다. 그림1.2에서 A 패킷의 도착 시간 지연 D1보다 B 패킷의 도착 시간 지연 D2가 작고 또 C패킷의 도착시간 지연인 D3보다는 D1이 작은 이런 현상이 발생한다.

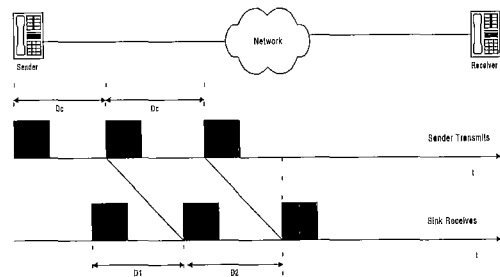


그림 1.2 패킷 도착시간 변이

이처럼 야기된 패킷 도착 시간 변이 현상을 없애기 위하여 기존에 사용하는 방식은 변이제거용 버퍼를 사용하고 있다^{[4],[5]}. 실제로 사용되는 RTP 프로토콜을 위한 RTP 헤더 모습을 그림1.3에서 보여준다. 패킷 도착 시간 변이 현상을 제거하려면 먼저 얼마나 변이가 일어났는가를 확인하여야 한다. 이 목적으로 사용하는 방법이 4바이트로 구성된 타임스탬프를 이용한다. 음성 패킷인 경우에 패킷들을 보낼 때 타임스탬프 값은 샘플링율에 따르는데 20msec 동안의 음성 정보를 가지는 패킷마다 $8000 \times 0.02 = 160$ 씩 증가하는 타임스탬프 값을 갖는다^{[6],[7]}.

음성이 없는 침묵 기간이 중간에 있어도 타임스탬프 값은 증가하게 되는데, 이는 패킷의 분실을 찾아내기 위하여 TCP 프로토콜에서 오류제어용으로 사용하는 그림1.3의 Sequence number 필드 값의 이용 방법과의 큰 차이점이다. 즉, Sequence number 필드 값은 각각의 RTP 패킷이 송출되면서 1씩 증가하게 되나 타임스탬프 값은 160 샘플 단위

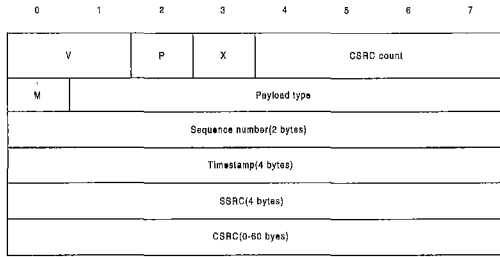


그림 1.3 RTP 헤더구조

로 패킷의 유무에 상관없이 계속 증가하게 되어 시간의 속성을 가지게 된다. RTP 패킷이 수신 측에 도달할 때 도착 시간 변이가 계산되는 방법은 수신 측이 보유하고 있는 타임스탬프 단위 즉 1초에 8000씩 증가하는 국부 클럭에 의해서 도착 패킷의 헤더에 있는 타임스탬프 값과 비교하게된다. 결국 RTP 패킷 수신 지점에서 수행되는 도착 시간 변이를 없애는 방법은 타임스탬프 값을 보고 지연시간 균등화 방법으로 처리하는 것이다^{[5]-[7]}.

ITU-T G.114 권고안에는 VoIP 전화에 있어서 음성 패킷이 송신측에서 출발하여 수신측에 도달할 때까지의 시간 지연이 150msec 이내이어야만 만족할만한 음질을 보장할 수 있다고 권장하고 있다. 그러나 150msec 지연 시간 한계는 단순히 권장 사항이고 실제로는 110msec 정도로 고정 시간 지연을 고려할 때 이 가운데 약 60msec 정도는 지연시간 균등화를 위하여 변이 제거용 버퍼에서 패킷이 강제적으로 기다리는 시간이라고 볼 수 있다. 즉, 첫 번째 RTP 패킷이 수신측에 도달한 지점부터 변이 제거용 버퍼에서 60msec을 기다리게 된다. 이 60msec 동안 평균적으로 3개의 RTP 패킷이 변이 제거용 버퍼에서 대기하게되고 국부클럭에 동기시켜 대기중인 RTP 패킷의 헤더에 있는 타임 스탬프 값을 참고하여 변이 제거용 버퍼에서 출력되는 모양은 송신측에서 보낼때와 같은 모양의 지연시간 변이 성분이 사라진 RTP 패킷 흐름으로 변환되게 된다. 여기서 얼마의 크기로 발생할지 모르는 IP 네트워크를 통한 변이 시간을 없앨 목적으로 어떠한 경우든지 60msec의 추가 시간 지연이 강제적으로 발생한다는 점은 타임스탬프에 의한 지연시간 균등화 방법에 입각한 RTP의 단점이다^[6]. 이는 IP 네트워크의 상태가 아주 좋아서 송신측에서 송출할 때의 균등한 패킷 흐름과 같은 모양으로 수신측에 도달했다고 할 경우에도 강제적으로 60msec의 시간 지연이 추가되어 결국은 변이 제거용 버퍼의 사용이

성능을 떨어뜨린다는 점이다. 이와 같이 현재 RTP 방식은 나쁜 조건 한계(pessimistic bound)에 입각하여 시간 지연을 보상하는 기법이기 때문에 이 논문에서는 IP 네트워크의 상태에 적응하여 동작하기 때문에 강제적 시간 지연이 필요없는 적응식 기법을 제시하고 이를 분석해 본다.

II. 적응식 웨이핑 방법 제안 및 성능 분석 모델

2.1 동적으로 패킷 도착시간 변이를 제거하는 적응식 웨이핑 방법 제안

기존에 사용되고 있는 타임스탬프를 이용한 지연 시간 균등화 RTP 패킷 전송기술은 IP 네트워크의 상태에 따라 동적으로 적응하지 못한다는 점에 착안하여 이 논문에서 IP 네트워크의 상태에 동적으로 적응하는 적응식 변이 제거 방식의 RTP 패킷 전송 기술을 제안한다.

먼저 이 논문에서 제안하는 방법에서도 송신측에서는 기존에 RTP 음성 패킷 전송시에 사용하는 타임스탬프 필드를 그대로 이용하고 또 타임스탬프 값을 패킷마다 설정하는 방법도 샘플링율에 근거한 시간, 즉 1/8000 시간 주기가 지남에 따라 타임스탬프 값이 1씩 증가하는 것으로 한다. 그림2.1과 같이 기존 방식과 마찬가지로 음성이 없는 구간(silence 구간)에서도 타임스탬프 값은 증가하게 되어 RTP 패킷들의 순서를 강조하는 순서 번호부여 개념이 아니고 시간의 상대적 흐름에서의 패킷의 위치를 표현해 주는 시간개념으로 타임스탬프 값이 부여된다.

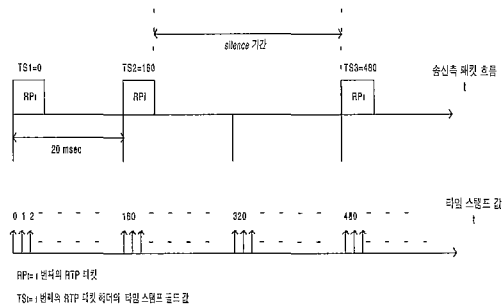


그림 2.1 송신측에서의 타임스탬프 설정

그림2.1에서 송신측에서 RTP 패킷을 보낼 때 순차번호가 (SN)가 SN_i 인 i 번째 RTP 패킷을 RP_i 라 부르고 i 번째 RTP 패킷의 헤더에 있는 타임스탬프 필드 값을 TS_i 라 표현한다. 이때 송신측에서

출발하는 패킷 흐름에서는 20msec 단위로 160번의 샘플값이 하나의 인접 패킷들간의 시간 간격에 대응하므로

$$TS_{i+1} - TS_i \geq 160 \quad (1)$$

의 관계가 성립한다. 여기서 $TS_{i+1} - TS_i = 160$ 인 경우는 RP_{i+1} 과 RP_i 가 Talk 기간의 인접 패킷인 경우이고 $TS_{i+1} - TS_i > 160$ 인 경우는 RP_{i+1} 과 RP_i 패킷이 silence 기간으로 떨어져 있는 경우이다.

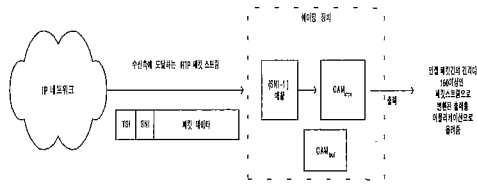


그림 2.2 셰이핑 장치

그림 2.2에 표시된 셰이핑 장치의 역할은 수신한 트의 IP 계층에 도달된 RTP 패킷 스트림을 분리해 내어 RP_i 의 헤더에 있는 TS_i 값을 조사한다. 여기서 만약 RP_{i-1} 이 셰이핑 장치를 떠난 출력된 시간을 DT_{i-1} 이라고 하고 RP_i 가 셰이핑 장치에 도달한 시점을 AT_i 라 정의하자. 그러면 현재 도착한 RP_i 의 TS_i 값을 조사한 후 다음의 관계식

$$AT_i - DT_{i-1} \geq \text{MIN}(160, (TS_i - TS_{i-1})) \quad (2)$$

을 만족하면 RP_i 를 바로 출력으로 내보내고

$$DT_i = AT_i \quad (3)$$

가 된다. 만약 (2)식을 만족하지 못하면 바로 앞서 들어온 인접 패킷과 현재 들어온 패킷과의 거리가 160이 되지 않는 경우이고 이 상태가 패킷도착 시간에 변이가 일어난 시점이다.

즉,

$$AT_i - DT_{i-1} < 160 \quad (4)$$

을 만족하는 경우가 발생하면 뒤에 들어온 패킷 RP_i 를 RP_{i-1} 이 셰이핑 장치를 출발한 시점부터 샘플링 클럭이 160이 경과한 시점까지는 CAM_{buf} 에 잠시 저장해 두어야 한다.

이 논문에서 제안하는 셰이핑 방법의 핵심인 이 CAM_{shape} 와 CAM_{buf} 의 동작원리는 다음과 같다. 먼저 CAM_{shape} 에 기억되는 각각의 저장 엔트리에는 현재 CAM_{buf} 에 잠시 저장되어 셰이핑 장치로부터 출발이 지연되고 있는 패킷들의 정보가 그림 2.3과 같이 기록되어 있다.

Sequence Number	COUNTER	TIMER	패킷 데이터
SN _i	0	160	NULL
⋮	⋮	⋮	⋮

그림 2.3 CAMshape 데이터 저장 구조

이 그림에서 RP_i 패킷의 순차번호(Sequence Number)값은 SN_i 로 표현된다. 이 값은 그림 1.3에서 헤더 필드에 있는 Sequence Number값이며 송신측에서 RTP 패킷 흐름에 대하여 각각의 패킷에 순차적으로 부여한 값이다. 또 이 CAM_{shape} 에는 BUF_{max} 개만큼의 엔트리를 보유할 수 있어 결국 BUF_{max} 개의 RP 패킷이 패킷 전송중에 발생한 전송 지연 시간 변이 성분을 없애기 위하여 셰이핑 장치에 저장될 수 있다.

우선 RP_i 가 들어오면 ($SN_i - 1$)번의 Sequence Number를 가진 엔트리가 CAM_{shape} 에 존재하는지를 확인한다. 이때 CAM_{shape} 의 Sequence Number필드 값들 중에서 ($SN_i - 1$)의 매칭이 일어나지 않으면 바로 앞저간 RP_{i-1} 과 RP_i 와의 관계가 (2)식을 만족하는 경우이다. 그러므로 이때는 CAM_{shape} 에 엔트리가 하나도 없는 경우에 해당되므로 RP_i 를 바로 셰이핑 장치 출력으로 내보내고 다음에 들어오는 패킷이 검토할 수 있도록 엔트리를 하나 CAM_{shape} 에 저장한다. 이 때 저장되는 엔트리 값은 그림 2.4와 같다. 이 경우의 엔트리 정보는 단지 다음에 들어올 패킷과의 샘플링 클럭 간격만을 측정하기 위한 것이므로 RP_i 의 패킷 데이터 부분을 저장할 필요가 없다.

Sequence Number	COUNTER	TIMER	패킷 데이터
SN _i	0	160	NULL

그림 2.4 CAMshape가 빈 상태에서 신규 엔트리 생성

그림 2.4에서 COUNTER값은 신규로 엔트리가 만들어지는 시점을 기준으로 CAM_{shape} 에 이미 저장되어 있는 RTP 패킷의 숫자를 의미한다. 그리고 TIMER 값은 엔트리가 처음으로 만들어질 때 160으로 초기화된다. 그리고 어떤 엔트리의 TIMER 값 TIM_i 는 국부 클럭에 따라 1초에 8000번의 시간단위로 1씩 감소하게 된다.

다음으로 만약 RP_{i+1} 패킷이 셰이핑 장치에 도달했을 때 $AT_{i+1} - DT_i < 160$ 인 경우가 발생한 경우는 그림 2.5와 같이 CAM_{shape} 에 RP_{i+1} 엔트리가 만들어진다. 즉 RP_{i+1} 이 도착하면 CAM_{shape} 와 $(SN_{i+1}-1)$ 값의 매칭을 시도할 것이다. 이때 CAM_{shape} 에는 $(SN_{i+1}-1) = SN_i$ Sequence Number 필드를 갖는 엔트리가 이미 존재하기 때문에 (이 경우는 $0 < TIM_i < 160$) 그림 2.5와 같은 엔트리가 하나 만들어지게 된다.

Sequence Number	COUNTER	TIMER	패킷 데이터
SN _i	0	TIM _i	NULL
SN _{i+1}	1	160 - TIM _i	RP _{i+1} 데이터

그림 2.5 CAMshape에 패킷 RP_{i+1} 저장 시점의 엔트리 생성

마찬가지로 샘플링 국부 클럭에 따라 각 TIMER 필드 값은 1씩 계속 감소한다. 그림 2.5인 상태에서 여기서 두 가지 경우를 생각해 볼 수 있는데 SN_i 엔트리의 TIM_i 값이 0이 되고 SN_{i+1} 엔트리의 TIMER 값이 160이 될 때까지 RP_{i+2} 패킷이 도착하지 않는 경우는 TIM_i 가 0이 되는 순간 SN_i 엔트리는 CAM_{shape} 에서 제거되고 SN_{i+1} 엔트리는 COUNTER 값이 0으로 바뀌고, TIMER 값이 160이 되며, 동시에 저장되어 있던 RP_{i+1} 패킷 데이터가 CAM_{shape} 로부터 출력되며, SN_{i+1} 엔트리의 패킷 데이터 필드가 NULL이 된다. 다른 한 경우는 TIM_i 값이 0이 되기 전에 RP_{i+2} 패킷이 셰이핑 장치에 도달한 경우는 순차번호가 $(SN_{i+2}-1)$ 을 CAM_{shape} 에 매칭시켰을 때 매칭이 일어난 경우이다. 즉, $(SN_{i+2}-1)$ Sequence Number를 갖는 엔트리 SN_{i+1} 엔트리가 CAM_{shape} 에 존재하고 있는 경우이다. 이때는 SN_{i+2} 엔트리가 하나 만들어지게 되는데 결과적으로 그림 2.6과 같이 바뀐다.

Sequence Number	COUNTER	TIMER	패킷 데이터
SN _i	0	TIM _i	NULL
SN _{i+1}	1	160 - TIM _i	RP _{i+1} 데이터
SN _{i+2}	2	320 - TIM _i	RP _{i+2} 데이터

그림 2.6 RP_{i+2} 패킷 데이터가 CAMshape에 저장

시간이 경과함에 따라 모든 엔트리의 TIMER 필드 값이 계속 감소하여 TIMER 값이 0이 되는 엔트리는 제거되고 또 TIMER 필드 값이 160이 되는 엔트리는 counter 값이 0으로 바뀌고, 저장된 RP 데이터가 출력되어 패킷 데이터 필드가 NULL이 된다. 또 TIMER 값이 320, 480, 640, ...에 이르는 엔트리는 각각 이 순간 COUNTER 값만 현재의 값에서 1씩 감소하게 된다.

2.2 셰이핑 장치 수행 특성 분석 모델

CAM_{shape} 에는 BUF_{max} 개의 엔트리가 저장될 수 있다. 만약 CAM_{shape} 에 BUF_{max} 개의 엔트리가 생성되어 있는 상태에서 도착하는 패킷은 패킷 손실로 간주한다. 분석모델이 Discrete Time Space 상에서 동작하는 것으로 가정하며, 이때 단위 시간 슬롯을 5msec로 설정하였다. 즉, 실제 시스템에서 40번의 샘플링 시간이 분석 모델에서 1시간 슬롯으로 동작하게 된다. 즉, 1초를 200개의 타임 슬롯으로 하나의 타임슬롯이 5msec인 Discrete Time Space로 분석모델이 이루어진다.

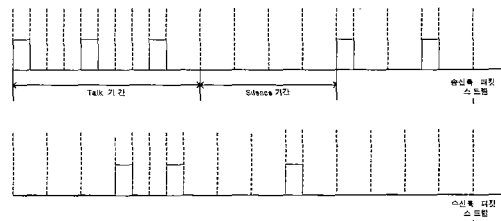


그림 2.7 CAMshape에 도달하는 패킷 스트림

그림 2.7을 보면, 송신측 패킷 스트림은 음성 트랙의 특성상 Talk 기간과 Silence 기간의 반복으로 스트림이 구성된다. Silence 기간에는 4개의 타임슬롯마다(즉, 샘플링율로 160번) RTP 음성 패킷이 생산되지 않는다. 여기서 송신측 발생 RTP 패킷 스트림의 특성 가운데 먼저 [평균 Talk 기간 / (평균 Talk 기간 + 평균 Silence 기간)] 값을 $TALK_{ratio}$ 로

정의한다. 또 송신시에는 Talk기간 동안 4개의 시간 슬롯마다 패킷이 발생한다. CAM_{shape} 에 도달하는 수신측 패킷 스트림 측면에서 관찰하면 RP_i 패킷이 도달하면 그림 2.2에서 이미 설명한 바와 같이 자기보다 바로 앞서 온 패킷의 Sequence Number 즉 SN_{i-1} 번호를 가지고 CAM_{shape} 에 매칭을 시도한다. 이때 랜덤변수 C 를 다음과 같이 정의한다.

$$\begin{aligned}
 & SN_{i-1} \text{과 매칭이 일어날 때,} \\
 & C = \left(\text{INTEGER} \left[\frac{TIM_{i-1}}{40} \right] + 1 \right) \\
 & SN_{i-1} \text{과 매칭이 일어나지 않을 때} \\
 & C = 0
 \end{aligned} \tag{5}$$

결국 C 값은 바로 앞서 CAM_{shape} 로 들어온 셀의 TIMER 필드 값을 의미한다. 여기서 임의의 n 번째 타임 슬롯에서의 C 값을 $C(n)$ 이라 표현하면 $C(n)$ 은 그림 2.8과 같이 변하는 확률과정이다.

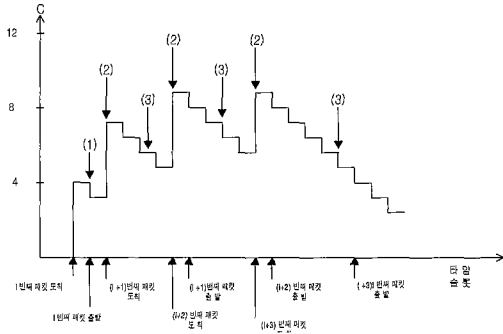


그림 2.8 확률 과정 $C(n)$

그림 2.8에서 C 의 값이 0인 곳은 CAM_{shape} 에 엔트리가 없는 경우이며, C 의 값이 0에서 출발하여 4로 갔다가 바로 3으로 바뀌는 (1)과 같은 패턴은 CAM_{shape} 에서 SN_{i-1} 매칭이 일어나지 않은 경우이다. 즉, RP_i 와 바로 앞서 CAM_{shape} 로 입력된 RP_{i-1} 과의 간격이 4이상인 상태를 의미한다. 이 경우 RP_i 는 그대로 CAM_{shape} 에서 출력된다.

C 의 값이 0이 아닌 상태에서 4만큼 점프하는 (2)번과 같은 경우는 RP_{i+1} 이 셰이핑 장치에 들어올 때 바로 전에 들어온 RP_i 가 셰이핑 장치를 떠나지 못하고 있거나 떠난 경우라 하더라도 아직 그 패킷이 떠난 후 4만큼의 시간이 경과하지 않은 경우이다. 이 때 입력된 RP_{i+1} 은 바로 셰이핑 장치에서

출력될 수 없고, CAM_{shape} 에 저장되었다가 앞서 들어온 RP_i 가 출력되고 4슬롯이 경과된 후에 셰이핑 장치에서 출력되어야 한다. 막 (3)번과 같은 모양은 C 의 값이 계속해서 1씩 떨어지는 부분의 슬롯에 셰이핑 장치로 패킷이 들어오지 않고 있는 경우이다. C 의 값이 4보다 큰 경우는 CAM_{shape} 에 저장 패킷의 엔트리가 존재하는 상태에서 CAM_{shape} 에 패킷 데이터가 1개 이상 저장되어 있는 경우이고, C 의 값이 0보다 크고 4보다 작은 경우는 CAM_{shape} 에는 패킷 데이터가 저장되어 있지 않으나 하나의 엔트리만 존재하는 그림 2.4와 같은 경우이다.

$C(n)$ 확률과정을 수학적으로 해석하기 위하여 CAM_{shape} 에 도달하는 $C(n)$ 을 베르누이 도착 과정이라고 가정한다. 이 베르누이 도착 과정의 가정은 도착시간 변이를 분석하는데 있어 최악 한계 (pessimistic bound)가정이다. 여기서 음미할 수 있는 것은 이 분석 방법으로 나온 결과 보다 실제 시스템이 좋은 성능을 보장한다는 점이다. 이때 임의의 타임 슬롯에 셰이핑 장치로 패킷이 들어올 확률은 p 는

$$p = \frac{TALKratio}{4} \tag{6}$$

이다.

그림 2.8의 확률과정 $C(n)$ 은 베르누이 도착과정의 가정에 기인하여 그림 2.9와 같이 상태 천이 다이어그램으로 나타낼수가 있다.

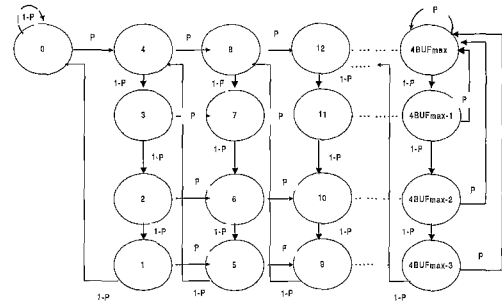


그림 2.9 확률 과정 $C(n)$ 의 상태천이 다이어그램

확률과정 $C(n)$ 은 상태공간이 $\{0, 1, 2, 3, 4, \dots, i, \dots, j, \dots, 4BUF_{max-1}, 4BUF_{max}\}$ 가 되는 마르코프 체인이며, 인접 슬롯에서 상태가 i 에서 j 로 바뀌어질 확률을 $P_{i,j}$ 로 표현하면 상태

천이 행렬 M 은

$$M = | P_{i,j} |,$$

$$i = 0, 1, 2, \dots, 4BUF_{\max} - 1,$$

$$j = 0, 1, 2, \dots, 4BUF_{\max} \quad (7)$$

이 되며

$$M = | P_{i,j} | = \begin{pmatrix} 1-p, & i=j=0 \\ 1-p, & j=i-1 \\ p, & j=i+4, & i \leq 4BUF_{\max} - 4 \\ p, & j=4BUF_{\max}, & i > 4BUF_{\max} - 4 \\ 0, & elsewhere \end{pmatrix} \quad (8)$$

이다.

평온상태에서의 C 의 확률값(steady state probability) 즉, $\lim_{n \rightarrow \infty} P\{C(n) = \theta\}$ 을 π_θ 로 정의하면, 열(ROW)벡터 $\Pi = (\pi_0, \pi_1, \pi_2, \dots, \pi_\theta, \dots, \pi_{4BUF_{\max}})$ 는 다음 관계식 $\Pi(I - M) = 0, \Pi e = 1$ 을 만족한다. 여기서 I 는 Identity Matrix이며 e 는 Unity Column Vector을 의미한다. Π 값의 의미는 RP패킷이 셰이핑 장치에 도착했을 때 CAM_{shape} 에서 기다려야 하는 지연시간의 분포를 의미한다. 임의의 패킷이 셰이핑 장치에 도착하여 CAM_{shape} 에서 기다려야 하는 지연시간을 확률변수 X 로 표현하면 $\pi_\theta = P\{X = \theta\}$ 의 관계가 성립한다. 이 때 셰이핑 장치에서의 평균지연시간을 T_{ave} 로 나타내면 확률과정에서 T_{ave} 의 관계식은 다음과 같다.

$$T_{ave} = \sum_{\theta=0}^{4BUF_{\max}} \theta \pi_\theta \quad (9)$$

셰이핑 장치에서의 셀 손실은 특정 RP패킷이 들어왔을 때 CAM_{shape} 가 가득 차 있기 때문에 이 특정 패킷이 버려지는 경우이다. 우선 특정 패킷이 셰이핑 장치로 들어올 때 CAM_{shape} 가 가득 차 있어서 버려질 확률을 P_{loss} 로 정의하면, 결국 이것은 CAM_{shape} 에 $4BUF_{\max}$ 개의 엔트리가 있을 확률과 같다. 즉, C 의 값이 $4BUF_{\max} - 3, 4BUF_{\max} - 2, 4BUF_{\max} - 1, 4BUF_{\max}$ 가운데 하나에 있을 확률값이다. 이를 식으로 나타내면 다음과 같다.

$$P_{loss} = \sum_{j=4BUF_{\max}-3}^{4BUF_{\max}} \pi_j \quad (10)$$

III. 성능 분석 결과

지연시간에 영향을 미치는 파라미터들은 식 (6)에서 p 값은 결정하는 $TALK_{ratio}$ 와 CAM_{shape} 의 최대 저장 엔트리 수인 BUF_{\max} 이다. 예를 들어 식 (6)에 의하면 셰이핑 장치에서의 패킷 손실율과 패킷지 $TALK_{ratio}$ 가 0.8인 경우 p 값은 0.2를 의미하며, 이는 임의의 타임 슬롯 안에 RP패킷이 도달할 확률이 0.2라는 것을 의미한다. 또 다음에 설명할 분석 결과는 앞에서 설명한 바와 같이 베르누이 패킷 도착과정을 가정한 분석 모델에 의한 결과이다. 이 모델이 갖는 의미는 IP 네트워크 상태가 좋지 않는 경우, 즉 도착시간 편이 현상이 크게 일어나는 네트워크 상태를 가정한 결과라고 볼 수 있다.

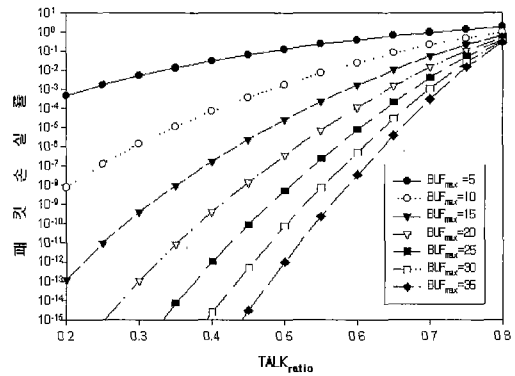


그림 3.1 셰이핑 장치에서의 패킷 손실률

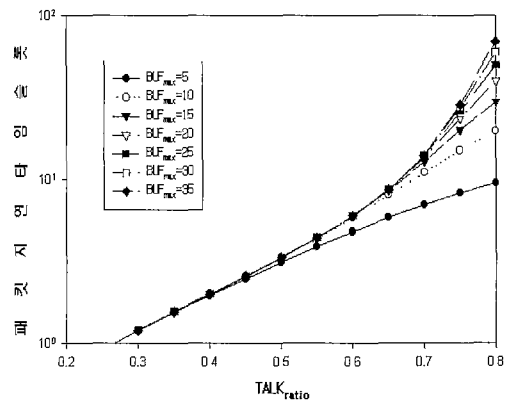


그림 3.2 셰이핑 장치에서의 패킷 지연 타임슬롯

그림 3.1은 셰이핑 버퍼 크기를 변수로 한 $TALK_{ratio}$ 에 대한 패킷 손실률을 보여주는 결과이

다. 이 그림에서 $BUF_{max}=20$ 개 일 때, 즉 인터넷을 통과하여 온 패킷들을 셰이핑할 수 있도록 20개의 패킷버퍼를 구비하면 $TALK_{ratio}$ 가 0.4인 조건에서 10-9 이상의 패킷 손실률을 보장받을 수가 있다. 여기서 $TALK_{ratio}$ 를 0.4로 두어 음성 데이터 전송 시간의 40%만이 실제 음성 데이터 전송이 일어남을 현실에 가깝게 가정하였고, 또 10-9의 패킷 손실률은 음성 통신에서 허용할수 있는 패킷 손실 수치로 보았다. $BUF_{max}=20$ 조건은 셰이핑 장치의 버퍼메모리가 $20*(160*8bit)=25.6Kbits$ 정도를 요구하는 수치이다. 이는 이 적응식 셰이핑 장치를 H/W적으로 구현해도 문제가 없음을 보여준다.

그림 3.2는 셰이핑 버퍼 크기를 파라미터로 하고 $TALK_{ratio}$ 대 패킷 지연시간을 보여준다. 그림 3.1의 결과에서 유도된 조건인 $BUF_{max}=20$ 이고 $TALK_{ratio}$ 가 0.4인 상태에서 2 타임슬롯의 시간 지연이 발생함을 알 수 있다. 이것은 실제 평균 지연 시간이 $2\text{타임슬롯} \times 40 \times 125\mu\text{sec} = 10\text{ msec}$ 가 된다. 여기서 이 논문에서 제안하는 방식의 장점을 보여주는데, 기존의 지연시간 균등화 방법은 약 60msec 동안 변이 제거용 버퍼에서 강제로 기다리게 하고 있다. 이 논문에서 제안하는 방법에 의하면 도착 시간 변이 현상을 없애기 위해 10msec만 추가 시간 지연이 있으면 된다.

IV. 결론

VoIP 전화에서 폰 음질에 영향을 미치는 패킷간 도착 시간 간격 변이 현상을 제거하기 위한 기존의 방법은 RTP프로토콜에서 채택하고 있는 패킷 지연 시간 균등화 방법이다. 이 방법은 처음부터 들어오는 패킷을 버퍼에서 강제 지연시켜서 그 뒤에 들어오는 패킷들이 버퍼에서 지연되면서 도착 시간 변이가 제거되는 방식으로 상태가 좋지 않은 IP네트워크를 고려하여 설계된 음질을 하향 평준화 시킨 프로토콜이다. 본 논문에서는 셰이핑 기술을 이용하여 IP네트워크의 상태에 따라 실시간으로 또 동적으로 도착 시간 변이 현상을 없애주는 적응식 변이 제거 셰이핑 방식을 제안했다.

기존의 방식과 비교할 때 적응식 셰이핑 장치는 IP네트워크의 상태가 좋아서 RTP 송신 스트림과 거의 같은 형태의 스트림이 수신측에 도달하면 셰

이핑 장치에서 지연시간 없이 그대로 bypass로 출력된다. 반면에 IP네트워크의 상태가 좋지 않아 도착 시간 변이 현상이 많이 일어나기 시작하면 변이 현상이 일어나는 시점에서만 동적으로 셰이핑 기능이 작동되는 네트워크의 상태에 적응하고 또 변이 현상이 일어나는 시점에 바로 대응하여 변이 현상을 제거해 주는 시스템이다.

제안하는 장치의 성능을 분석하기 위하여 수학적 으로 분석 모델을 개발하여 평균 TALK 기간 비율, 셰이핑 장치의 엔트리 저장 수 등을 파라미터로 하여 패킷 손실율과 셰이핑 장치에서의 평균 패킷 지연 시간을 분석하였다. 평균 TALK 기간 비율을 현실적인 값인 0.4정도로 보았을 때 셰이핑 장치의 저장 엔트리 수가 20개(25.6Kbits) 정도면 10^{-9} 정도의 패킷 손실율을 보인다는 사실을 확인하였다. 이 경우에 셰이핑 장치에서의 평균 지연 시간이 10msec라는 사실도 분석되었다. 결론적으로 본 논문에서 제안하는 적응식 셰이핑 방식을 이용하면 수신측 도착 시간 변이를 없애기 위해 평균 10msec 정도의 지연 시간만 고려하면 된다는 점이다. 이는 기존의 방식이 강제적으로 60msec 정도의 지연 시간이 강요되는 것과 비교할 때 성능이 훨씬 우수함을 알 수 있었다. 더구나 본 제안 방식은 IP네트워크의 상태가 좋은 경우에는 수신된 RTP 패킷 스트림이 셰이핑 장치를 거의 시간 지연 없이 bypass 되어 출력될 수 있다는 장점을 지니고 있다.

앞으로 계속될 확장 연구는 IPv6에서 실시간 흐름 트래픽을 레이블링하여 차별 처리할 때에 본 논문에서 제안한 적응 식 셰이핑 장치를 어떻게 적용시킬 것인가 하는 분야이다.

참고 문헌

- [1] Jonathan Davidson, James Peters, "Voice over IP Fundamentals", cisco systems, 2000
- [2] 강민규, 공상환,김동규,RTP/RTCP를 이용한 영상회의 시스템에서 오디오 패킷 손실보상을 위한 동적 부가 전송 매커니즘 개발 및 성능 분석", 한국정보처리학회 논문지, 1998.10
- [3] 고대식, "Real-time Transport Protocol(RTP)를 이용한 실시간 데이터 전송", 한국통신 학회지, 1997.6
- [4] Eric A. Hall, "Internet Core Protocols", O'REILLY, 1999
- [5] Walter J. Goralski & Matthew C.kolon, "IP

